

NumPy and Scipy

NumPy

- NumPy, short for Numerical Python is an open source Python library for scientific computing.
- It is one of the most important foundational packages for numerical computing in Python
- Most computational packages providing scientific functionality use NumPy's array objects for data exchange
- The NumPy package provides basic routines for manipulating large arrays and matrices of numeric data.
- It is a foundational package on which many of the data science packages are built.

NumPy

- While NumPy by itself does not provide modeling or scientific functionality, having an understanding of NumPy array and array-oriented computing will help you use tools with array-oriented semantic, like pandas, much more effectively.
- pandas builds upon functionality provided by Numpy.
- The pandas library relies heavily on the NumPy array for the implementation of the pandas Series and DataFrame objects, and shares many of its features such as being able to slice elements and perform vectorized operations.
- It is therefore useful to spend time going over NumPy arrays before diving into pandas.

NumPy

- For any scientific project, NumPy is the tool to know.
- It has been built to work with the N-dimensional array, linear algebra, random number, Fourier transform, etc.

The key features of NumPy are:

- `ndarrays`: n-dimensional arrays of the same data type which are fast and space-efficient. There are a number of built-in methods for `ndarrays` which allow for rapid processing of data without using loops (e.g., compute the mean).
- `Broadcasting`: a useful tool which defines implicit behavior between multi-dimensional arrays of different sizes.

NumPy

- Vectorization: enables numeric operations on ndarrays.
- input/Output: simplifies reading and writing of data from/to file.
- A C API for connecting NumPy with libraries written in C, C++, or FORTRAN

Why use NumPy?

- One of the reasons NumPy is so important for numerical computations in Python is because it is designed for efficiency on large arrays of data. There are number of reasons for this:
- NumPy is memory efficient, i.e., it can handle large amount of data more accessible than any other library.
- Moreover, Numpy is very convenient to work with, especially for matrix multiplication and reshaping as mentioned earlier.
- It is fast.

Why use NumPy?

- NumPy operations perform complex computations on entire arrays without the need for Python *for* loops
- Many advanced Python libraries, such as Scikit-Learn, Scipy, and Keras, make extensive use of the NumPy library.
- Therefore, if you plan to pursue a career in data science or machine learning, NumPy is a very good tool to master.
- So the knowledge of the NumPy library is a prerequisite in order to face, in the best way, all scientific Python packages, and particularly, to use and understand more about pandas library and how to get the most out of it.

NumPy

- For most data analysis applications, the main areas of functionality we'll focus on are:
 - Fast vectorized array operations for data munging and cleaning, subsetting and filtering, transformation etc.
 - Common array operations like sorting, unique, and set operations
 - Efficient descriptive statistics and aggregating/summarizing data
 - Data alignment and relational data manipulations for merging and joining together heterogeneous data sets
 - Group-wise data manipulations

SciPy

The SciPy (Scientific Python) package extends the functionality of NumPy with a substantial collection of useful algorithms, like minimization, Fourier transformation, regression, and other applied mathematical techniques.